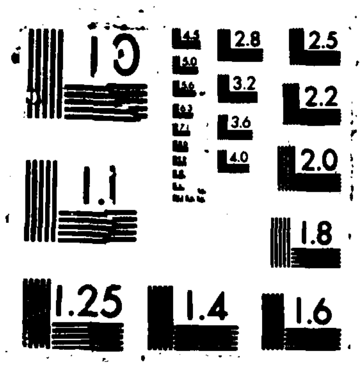
AD-A182 740 HUMAN CONFLICT RESOLUTION(U) WATERLOO UNIV (ONTARIO) RESEARCH INST N M FRASER ET AL JAN 87 AAMRL-TR-87-013 F33615-85-D-0514 1/1

UNCLASSIFIED F/G 12/9 NL

END
8-87
DTIC


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

AD-A182 740

(12)

AAMRL-TR-87-013

DTIC FILE COPY

HUMAN CONFLICT RESOLUTION (U)

NIALL M. FRASER
KEITH W. HIPEL
D. MARC KILGOUR

WATERLOO RESEARCH INSTITUTE

JANUARY 1987

Approved for public release; distribution is unlimited.

HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY
AEROSPACE MEDICAL DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

87 7 [illegible] 042

NOTICES

TECHNICAL REVIEW AND APPROVAL

AAMRL-TR-87-013

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

CHARLES BATES, JR.
Director, Human Engineering Division
Armstrong Aerospace Medical Research Laboratory

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AAMRL-TR-87-013

6a. NAME OF PERFORMING ORGANIZATION
Waterloo Research Institute

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Harry G. Armstrong Aerospace Medical Research Laboratory/HEC

6c. ADDRESS *(City, State, and ZIP Code)*
University of Waterloo
Waterloo, Ontario, Canada N2L3G1

7b. ADDRESS *(City, State, and ZIP Code)*
Wright-Patterson AFB OH 45433-6573

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F33615-85-D-0514

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
62202F	6893	04	D2

11. TITLE *(Include Security Classification)*
Human Conflict Resolution (U)

12. PERSONAL AUTHOR(S)
Fraser, Niall M., Hipel, Keith W., and Kilgour, D. Marc

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM APR 86 TO NOV 86

14. DATE OF REPORT *(Year, Month, Day)*
1987 January

15. PAGE COUNT
40

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
05	08	
09	04	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Conflict Resolution, Artificial Intelligence, AI, Decision Support, Methodology, Expert Systems, Knowledge-Based Systems, Cooperating Expert Systems.

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

A conflict resolver is developed for mediating disputes arising among competing expert systems. The conflict resolver constitutes the key component of an overall decision support system designed for operating and controlling a complex system. When competing expert systems make conflicting recommendations for the control of the system, the conflict resolver follows a Delphi-like approach in mediation to arrive at a compromise solution. Besides defining the mathematical framework for the comprehensive conflict resolver, it is proven rigorously that the conflict resolver always reaches a consistent recommendation.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Capt Daniel E. Snyder

22b. TELEPHONE *(Include Area Code)*
(513) 255-7590

22c. OFFICE SYMBOL
AAMRL/HEC

DD Form 1473, JUN 86 *Previous editions are obsolete.*

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

PREFACE

This study was funded and initiated by the Harry G. Armstrong Aerospace Medical Research Laboratory, Human Engineering Division, Wright-Patterson Air Force Base, Ohio. The bidding and co-ordination of the project was managed by Dr. Warren D. Peele of the Southeastern Center for Electrical Engineering Education (SCEEE), 1101 Massachusetts Avenue, St. Cloud, Florida 32769. The work was executed by Drs. Niall M. Fraser, Keith W. Hipel and D. Marc Kilgour at the Waterloo Research Institute, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1. The contract manager within the Waterloo Research Institute was Mr. David H. Copp. The research contract is entitled "Human Conflict Resolution" (Contract number F-33615-85-D-0514) and was completed during the time period from early April, 1986, to November 13, 1986.

During the execution of the research, the authors received valuable guidance and advice from Captain Daniel E. Snyder and Mr. Michael D. McNeese. Their assistance, as well as support from other personnel at the Wright-Patterson Air Force Base, is greatly appreciated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Within the general field of artificial intelligence (AI), there has been a recent explosion in the development of expert systems (ESs). Because there is a wide variety of complex problems in many different fields with the potential to be solved using ESs, the widespread development and use of ESs is imminent. For example, in the field of military science, the current and projected production of ESs has already been widely recognized.

To operate a given complex system, ESs covering different but related knowledge areas will have to be developed. Consider, for example, the case of a military aircraft. ESs have to be designed for handling a range of competing activities such as navigation, combat, flight engineering, and safety of the pilot. The employment of ESs and other knowledge sources having overlapping domains makes conflict inevitable among ESs that operate the system. The ESs within a system may suggest inconsistent (conflicting) recommendations that must be properly handled so that an overall consistent recommendation can eventually be made under operating conditions. The major objective of this project is to develop a procedure for mediating among ESs so that conflicts can be properly resolved. The specific mechanism for organizing and mediating conflict among ESs, as well as other knowledge sources, is referred to as the "conflict resolver." The conflict resolver, along with the ESs and other knowledge sources, forms a comprehensive decision support system for optimally operating the system.

In the next section, the literature related to the aforementioned objective is reviewed and appraised. Although the literature review clearly points out the need for the development of a conflict resolver, unfortunately no existing theoretical work on this topic has been discovered. Consequently, subsequent to the literature review, the theoretical framework for a conflict resolver is developed; it is proven mathematically that an overall resolution is always reached by the proposed approach. In order to make these new advances a practical reality, recommendations for future work are then presented.

2. LITERATURE REVIEW

The main purposes of the background review are to put the literature surrounding the development of the conflict resolver into proper focus and to point out some basic ideas that are used in the design of the conflict resolver in later sections. Following a general introduction to AI, expert systems are discussed. Within the overall structure of a decision support system, ESs can be used to operate and control a given complex system. Although ideas from cognitive psychology can be useful for basic research in AI and ESs, concepts from other areas are needed to assist in designing a conflict resolver for mediating disputes among ESs. Of particular importance is the Delphi approach to mediation. Because the game theory literature does not have a suitable methodology for resolving conflicts among ESs, the need for the theoretical development of a conflict resolver is indicated. Within the military sciences, the wide applicability of a general conflict resolver is emphasized.

2.1 Artificial Intelligence

Because AI is such a new and dynamic field of scientific research that encompasses a wide variety of topics, no concise and widely accepted definition of AI is currently available. Nevertheless, by studying alternative definitions of AI one can appreciate the general ideas behind the field, what types of problems AI is designed to solve, and in what directions AI is heading. Some of the definitions of AI include:

1. AI is the study of ideas that allow computers to be intelligent (Winston, 1984).
2. AI is the study of how to make computers do things at which, at the moment, people are better (Rich, 1983).
3. AI is that part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behavior (Barr and Feigenbaum, 1982).
4. AI is that branch of computer science dealing with symbolic, nonalgorithmic methods of problem solving (Buchanan and Shortliffe, 1984).
5. AI is the branch of computer science that deals with ways of representing knowledge using symbols rather than numbers and with rules-of-thumb, or heuristic, methods for processing information (Buchanan, 1985).

Mishkoff (1985) compares the foregoing definitions of AI and explains how each of them is useful for understanding what AI is about. Because AI research involves using computers to simulate human intelligence, some researchers prefer to use the term "machine intelligence" instead of AI.

Although concepts related to AI can be traced back for centuries, almost all of the major developments in AI took place after World War II. In 1956 at Dartmouth College in Hanover, New Hampshire, researchers attempting to simulate human intelligence on a computer held a conference to discuss their ideas. Many of the major American pioneers in AI attended the Dartmouth Conference and the term AI was coined at the conference. For example, Dr. John McCarthy, who in 1958 invented the now popular AI programming language LISP (an acronym derived from LISt Processor), was one of the conference organizers. As demonstrated by the founding of AI societies, establishment of AI journals, publication of numerous papers and books on AI, and widespread application of AI in many areas, the field of AI has flourished since 1956 and evolved into one of the most interesting and exciting fields of research of the "high technology" era. In many industrialized countries, AI societies have been formed, such as the American Association for Artificial Intelligence (AAAI) in the United States, and the Canadian Society for Computational Studies of Intelligence in Canada. A journal entirely devoted to publishing AI research is *Artificial Intelligence, an International Journal.* The Institute of Electronic and Electrical Engineers (IEEE) publishes as one of its many professional journals the *IEEE Transactions on Pattern Analysis and Machine Intelligence* which contains many AI papers. Original AI research also appears in systems engineering, operations research and computer science journals. For example, *IEEE Transactions on Systems, Man and Cybernetics,* and *Large Scale Systems* also have AI papers. Applications of AI within a given discipline such as mechanical engineering are often published in the journals representing that discipline. In fact, due to the importance of AI across engineering disciplines, a journal called *Artificial Intelligence in Engineering* has been started. Besides the AI textbooks cited with the definitions of AI at the start of this section, other books that can be referred to include texts by Sell (1985), Davis and Lenant (1982), Nilsson (1980), Charniak et al. (1980), Boden (1977) and McCorduck (1979). Special collections of papers which were initially published in journals, conference proceedings or technical reports, have also been published as books (see, for example, Feigenbaum and Feldman (1963), Schank and Colby (1973), Bobrow and Collins (1975), and Findler (1979)). Every three out of four years, and every other year, the AAAI and the International Joint Conference on Artificial Intelligence, respectively, hold conferences for which proceedings are published.

Because human intelligence is required in all of mankind's undertakings, AI has the potential of playing a major role in almost every discipline. For instance, ESs which are discussed in some detail in the next section, could be useful in disciplines ranging from medical science to car mechanics. Due to the proven and potential widespread applicability of AI in many different fields, there is a wide variety of research topics in AI. Some of these topics are listed in Table 1. Because ESs are of particular importance in military and other conflict situations, these are reviewed next.

Table 1: Topics in AI

Expert Systems
Game Playing (ex., chess and checkers)
Theorem Proving
Natural Language Processing (understanding and generation)
Speech Recognition (understanding and generation)
Computer Vision
Robotics
Intelligent Computer Aided Instruction
Software Development
Planning and Decision Support
Factory Automation
Office Automation
Other Application Areas

2.2 Expert Systems

Definition

According to Negoita (1985), ESs are software systems that mimic the deductive or inductive reasoning of a human expert. In a particular field, an ES can be used to assist an expert or provide information to a lay person who does not have access to an expert. Within the field of medicine, for example, ESs have been developed for diagnosing diseases (Buchanan and Shortliffe, 1984). ESs have also been used in domains such as intelligent computer-aided instruction for mathematics, oil exploration, job-shop scheduling, and planning experiments in molecular genetics (Mishkoff, 1985, Ch. 3). As a matter of fact, the development and application of ESs in different fields have caused AI to become well known and popular among both researchers and practitioners.

Components

The main components of most ESs are a knowledge base, an inference engine and a user interface. Because of the great importance of the knowledge base, ESs are commonly referred to as "knowledge-based systems." The construction of ESs is called "knowledge engineering."

A knowledge base contains two types of knowledge. "Declarative knowledge" consists of the facts about objects, events and situations while "procedural knowledge" is the information about courses of action. Depending upon the form of the knowledge representation that is used, the two kinds of knowledge may be separate or combined. The most common form of knowledge representation is the "rule-based production system." Another attractive approach is the "model-based" representation.

The inference engine determines how and when to use the information contained in the knowledge base. Because the inference engine runs an ES, it is also referred to as the "control structure" or the "rule interpreter." Due to the fact that an inference engine is independent of the knowledge base, an inference engine can often be used with different knowledge bases.

To permit communication with humans, a "user interface" is required with every ES. Because the intended users of ESs are usually computer neophytes, one must be able to communicate with ESs using ordinary English statements. Although bidirectional communication between an ES and user is now usually executed using written and graphical displays on a video display screen, in the long run researchers would like to develop facilities for verbal communication. The area of AI that deals with programming computers to understand and generate natural language is called "natural language processing."

In the future, complex systems may be modelled and controlled using multiple ESs as well as human beings. For example, in a military aircraft the pilot may have to interact with various ESs in the control of the functions of the aircraft. To allow the various knowledge sources such as ESs and human beings to communicate with one another, the concept of a *blackboard* can be utilized. The blackboard consists of a shared data structure to which all knowledge sources have access. When an ES interacts with a blackboard, it can use the blackboard to write down its own "output" or current state and also to obtain the output from other knowledge sources as "input" for further consideration. Recent research developments in blackboard architecture are presented by Hayes-Roth (1985).

Construction

The building of ESs falls within the realm of knowledge engineering. Knowledge engineers and domain experts form the two categories of people needed to construct ESs. A knowledge engineer is an AI specialist who is trained in developing an ES while a domain expert is a person who has professional training in the domain that is being developed as an ES. Based upon the knowledge that is passed to him by the domain expert, the knowledge engineer iteratively constructs an ES. As explained in detail by Hayes-Roth et al. (1983), the steps required in building an ES are identification, conceptualization, formalization, implementation and testing. Usually this is a very time consuming and expensive process.

Specialized computer languages have been developed for use in programming ESs as well as other AI applications. LISP may be the most popular AI language although PROLOG is now used extensively for AI applications. Initially, ESs were programmed directly using an AI language such as LISP. However, rather than develop each ES from scratch using LISP, developmental tools are now available. These tools consist of high level programs that allow ESs to be programmed relatively quickly and efficiently. Besides high level software, hardware is also designed and manufactured for use in AI applications such as the development of ESs. One particular type of hardware is the LISP machine which is designed primarily for the development of AI programs. A LISP machine typically has features that include a high speed processor, large memory, bit-mapped display, specialized keyboard and mouse.

2.3 Decision Support Systems

As was also the case for AI, there is no generally accepted definition for decision support systems (DSSs) (Sol, 1985). Ginzberg et al. (1982) explain how various definitions of DSSs have evolved over the years from the early 1970's to the present, as the DSS field has developed and expanded. To appreciate what DSSs are all about and what they are attempting to accomplish, it is again informative to examine alternative DSS definitions. Some of these definitions include:

1. Systems to support managerial decision makers involved in unstructured or semi-structured decision situations (Gorry and Scott Morton, 1971).

2. Systems to support the decisions of managers. DSSs allow managers to utilize the query capabilities of the computer to obtain requested information and retain control over the decision-making process as changes occur. Whereas classical management information systems focus on structured problem solving, DSSs extend the range of problem structure to include semi-structured and

unstructured problems for which interactive problem solving is required (Theirauf, 1982).

3. Extensible systems which are capable of supporting ad hoc data analysis and decision modelling, oriented towards future planning, and used at irregular, unplanned intervals (Moore and Chang, 1980).

4. Computer-based systems consisting of the three interactive components:

   i) a language system for providing communication between the user and other DSS components;

   ii) a knowledge system that contains the problem domain knowledge either as data or procedures;

   iii) a problem processing system that links the other components and possesses general problem manipulation capabilities required in decision making (Bonczek et al., 1980).

5. The use of computers to assist managers in their decision processes in semistructured tasks, support (rather than replace) managerial judgement, and improve the effectiveness of decision making rather than its efficiency (Keen and Scott Morton, 1978).

In addition to those given above, other definitions and discussions of DSSs are presented by authors such as Keen (1980), Sprague (1980), Bonczek et al. (1981), Sol (1985), and Klein and Hirscheim (1985). Original research regarding DSSs is published in operations research, management sciences, management information systems, and decision analysis journals. Due to the widespread demand for research into and application of DSSs, Elsevier Science Publishers launched the international journal *Decision Support Systems* in 1985. A variety of interesting articles on DSSs can be found in this journal. Dutta and Jain (1985), for example, develop DSSs for distributed computer system design in the presence of multiple conflicting objectives. Jarke (1986) explains how knowledge sharing and negotiation support can be carried out in multiperson DSSs. Sol (1985) addresses solutions to problems involving the aggregation and disaggregation of data in DSSs. Landry et al. (1985) define the term "problem" within the context of DSSs in order to improve the effectiveness and potential applications of DSSs.

As noted by Ginzberg et al. (1982), the area of DSSs attempts to bring together and focus a number of independent disciplines for the purpose of improving decision making in organizations. These disciplines include operations research, management science, data-base technology, systems engineering, decision analysis and AI. Traditionally, DSSs have provided support for decision makers but have not been able to devise original solutions to problems on their own. In order to develop DSSs that can "think," concepts from AI are currently being introduced into DSSs. Of particular importance is the incorporation of ideas from ESs into DSSs (Singh and Cook, 1985; Sen and Biswas, 1985).

Numerous applications of DSSs to a wide range of different kinds of problems are given in the published literature such as the texts by Keen and Scott Morton (1978) and Ginzberg et al. (1982). As discussed in detail below, a complex DSS may consist of an array of ESs that have conflicting objectives and a conflict resolver that can resolve disputes among the ESs. The ESs can communicate with one another using a blackboard and, as well, interact with the conflict resolver and a decision maker. An example of this type of DSS would be the Pilot's Associate in a combat aircraft.

2.4 Cognitive Psychology

The diverse field of cognitive psychology deals with developing theories of human intelligence. Research in cognitive psychology appears in a wide range of journals such as psychophysics, neuroscience, psycholinguistics, cognitive anthropology and education journals. Some of the best known journals that are mainly concerned with basic research in cognition include *Cognitive Science*, *Cognitive Psychology*, *Cognition*, *Behavioral and Brain Science*, *Journal of Experimental Psychology, Memory and Cognition,* and *Psychological Review.*

AI scientists are attempting to develop both software and hardware that will allow computers to "think like humans" in a variety of situations. Consequently, theories of human intelligence developed by cognitive researchers can be used by AI scientists for developing and testing AI theories. To inform AI researchers about current developments in cognitive psychology, review articles regarding aspects of cognitive psychology that could be useful in AI are sometimes published in AI journals. For example, Smith (1985) outlines recent advancements in areas of cognitive psychology such as prototype theory, inductive reasoning, and deductive reasoning. Other review articles about cognitive psychology are those by Pylyshyn (1982) and Anderson (1984). Rumelhart et al. (1986) derive a theoretical framework for describing parallel distributed processing and apply their framework to the development of models of cognition, perception, memory, language and thought. Their research ties together cognitive psychology, AI and neuropsychology in an integrative fashion.

2.5 Delphi Method

The Delphi method is an iterative procedure that anonymously uses the opinions of experts to arrive at a resolution to a problem (see, for example, Quade (1985, pp. 200-201)). At the first step, each expert is asked to respond to a written questionnaire which could appear on paper or on a computer monitor. In each of the rounds after the first, each expert is provided with the responses of all others. However, in order to avoid the psychological drawbacks associated with face-to-face committee meetings, the identities of the authors of the various opinions are not revealed. Furthermore, at a given round, an expert may be asked the reasons behind his previously expressed opinions and be asked to consider other possible approaches to solve the problem. As each expert revises his position over a series of rounds based upon the evolving opinions of the other experts, this anonymous debate may eventually lead to a consensus, or at least to a narrowing of the range of viewpoints.

The Delphi technique was originally developed at the Rand Corporation (Dalkey, 1969) and has been widely applied in many different fields. Most textbooks in operations research and decision analysis contain at least some discussion of the Delphi method and its applications. As explained in the technical section of this report, an "automated" Delphi method is designed below for permitting a conflict resolver to arrive at a consensus among competing ESs. It is proven mathematically that a consistent recommendation is always reached using this automated Delphi method.

2.6 Game Theory

A game is a model of a conflict where two or more groups with decision-making power are in dispute over some issue(s). Examples of conflicts which can be modelled as games include military campaigns, arms reduction negotiations, environmental impact assessments of engineering projects, parliamentary maneuvering, trading disputes, and even landlord-tenant controversies. Because conflict is an inherent characteristic of human behaviour, game models constitute valuable tools for understanding and at least partially controlling the real world.

Due to the great need for the study of conflict in many different fields, various game-theoretical methodologies have been developed for modelling disputes, forecasting their resolutions and suggesting routes for optimal decision making. A landmark development in the theory of games was the pioneering research of von Neumann and Morgenstern (1953), of which the first edition was published during World War II. Since the war, there has been a deluge of research regarding various aspects of game theory. Even though game theory is often thought of as a branch of other fields such as operations research or mathematics, it can rightfully be considered as a separate academic field on its own.

Because of the large amount of research published in game theory, it is necessary to categorize game theory research conveniently in order to appreciate what has been done and what remains to be done. Two specific branches of game theory are cooperative and noncooperative game theory. In cooperative game theory, the decision makers (who are also called players or participants) have decided to cooperate and the problem to be solved is how to divide the pie equitably among them all. A wide variety of theories have been developed for recommending how the joint resources can be fairly allocated. In terms of research activities, more effort has apparently been devoted to work in cooperative game theory as compared to noncooperative game theory. Besides von Neumann and Morgenstern (1953), another influential text that contains a significant amount of material regarding cooperative game theory is Luce and Raiffa (1957). Published papers on cooperative game theory appear in the *International Journal of Game Theory* as well as many operations research, mathematics and economics journals.

In noncooperative game theory, the decision makers act independently of one another and no decisions have been made regarding cooperation or other tactical and strategic matters. In other words, they have not decided to divide up the pie; the amount of the resource may still be at issue and, in particular, any one of the players may try to run off with the entire pie under his arm. Research papers on noncooperative game theory appear in systems engineering journals such as *IEEE Transactions on Systems, Man and Cybernetics,* and *Large Scale Systems,* in conflict resolution journals such as the *Journal of Conflict Resolution,* and *Conflict Management and Peace Science,* in operational research journals like *INFOR*, *OMEGA* and *The Journal of the Operational Research Society,* as well as in journals published by disciplines such as political science, international studies, environmental studies, and water resources. In a recent paper, Kilgour et al. (1984) mathematically compared a wide range of solution concepts (i.e. mathematical models of human behaviour) and put research in noncooperative game theory into proper perspective. Important textbooks on noncooperative game theory include Fraser and Hipel (1984) and Howard (1971).

Cooperation can be introduced into noncooperative game theory. One mechanism for accomplishing this is to employ a mediator who may assist or even enforce cooperation among competitors. In other words, a mediator can change the rules of play of the game.

With the advent of the field of AI and, in particular, the development of ESs within AI, game theorists have been confronted with a challenging new research problem. How can conflicts among ESs, contained within an overall system such as a combat aircraft, be properly resolved? Before the initiation of the current project, no theory existed for solving this formidable problem. However, based upon some key structural characteristics that appeared in earlier game theory research by the authors of this report, and others, a solution to this complex problem is developed later in this manuscript. More specifically, the conflict to be solved involves ESs which competitively interact with one another as well as human decision makers. The theoretical mathematical framework is designed for a "conflict resolver" that acts as a mediator to reach a solution to a conflict by following the Delphi approach described earlier. Furthermore, a comprehensive theorem proves that a unique solution will *always* be found for this type of conflict. By allowing distributed sources of knowledge from competing ESs to cooperate as a total knowledge concept through the use of the conflict resolver, game theory now has an important and established role to play within the blossoming field of AI in general, and ESs in particular.

2.7 Military Science

For a long time, game theory has been used to model, understand and solve military conflicts. Within the text of Fraser and Hipel (1984), for example, many military disputes are modelled using concepts from noncooperative game theory. Other military applications of game theory can be found in the journals referred to in the previous section.

Recently, the great potential application of AI within the field of military science has been recognized by a number of countries, especially the United States of America. For example, Anderson et al. (1985) explain how several mutually cooperating ESs can assist combat pilots. McNeese (1986) describes a human systems engineering approach for combining the best features of AI with human capabilities and limitations in order to create an intelligent cockpit that optimally operates an aircraft. Because SAINT (Systems Analysis of Integrated Networks of Tasks) furnishes the conceptual framework for representing systems that consist of discrete task elements, continuous state variables and interactions among them (Wortman et al., 1977), it could be useful for designing ESs that must perform specified tasks. Among other tools, Hoyland et al. (1985) use SAINT in a study of the incorporation of human operator considerations into the analysis of weapons delivery systems. Based on discussions at a workshop sponsored by the Army Research Institute, Sage and Rouse (1986) report on how human decision making can be enhanced by the knowledge-based sciences. One particularly challenging problem is to employ ESs to assist pilots flying combat aircraft. The overall system consisting of interacting ESs and a conflict resolver for resolving conflicts among the ESs, and perhaps also the pilot, is referred to as the "Pilot's Associate". Buchanan (1982) suggests that the future of ESs will be severely restrained by factors that include conflicts in plans, strategies and methods as well as inconsistencies in working with multiple sources of knowledge. Consequently, in order to make a system such as the Pilot's Associate operational, a crucial development is the theoretical design of the conflict resolver contained within the Pilot's Associate This major objective is successfully fulfilled in the next sections of this report.

3. A DECISION SUPPORT SYSTEM FOR CONFLICT RESOLUTION

3.1 Overview

Figure 1 illustrates an overall structuring of the conflict resolver problem. The two basic components are the Decision Maker (DM) and the Decision Support System (DSS). The key element of the DSS is the Conflict Resolver (CR).

The DM takes action in response to external directives and perceptions, and also takes into account interaction with the DSS. In the example of the future design of a combat aircraft, the DM is the pilot, while the DSS is the Pilot's Associate. External directives would include instructions from ground, and perceptions would incorporate both the readings of instruments and visual observations. The pilot can also make inquiries of the DSS through the CR, and limited inquiries with the external directive sources.

The DSS consists of a finite number, m, of Expert Systems (ESs) and the CR. The DSS senses the environment through the functions of the ESs. It uses the CR to resolve contradictory recommendations among the ESs and reports the overall recommendation to the DM. Autonomic actions as required can also be performed by the DSS, either via the CR or by the ESs directly.

The DM requires a single recommended set of actions at any one time. The function of the DSS is to develop a comprehensive recommendation by compiling information from the ESs, taking into account directives from the DM and from external sources.

There are several information types that appear in this structure:

1) directive - an instruction for action. This can come from the environment (eg. ground control) or from the DM to the CR.

2) informative - data concerning the environment. This is data that the ESs collect, or the perceptions of the DM.

3) supportive - a recommendation based upon information about the environment and specialized knowledge from the ESs to the CR, or from the CR to the DM. These differ from directives in that they can be used or not used as the receiver wishes. They are also supported by a query facility to permit adjustment or clarification.

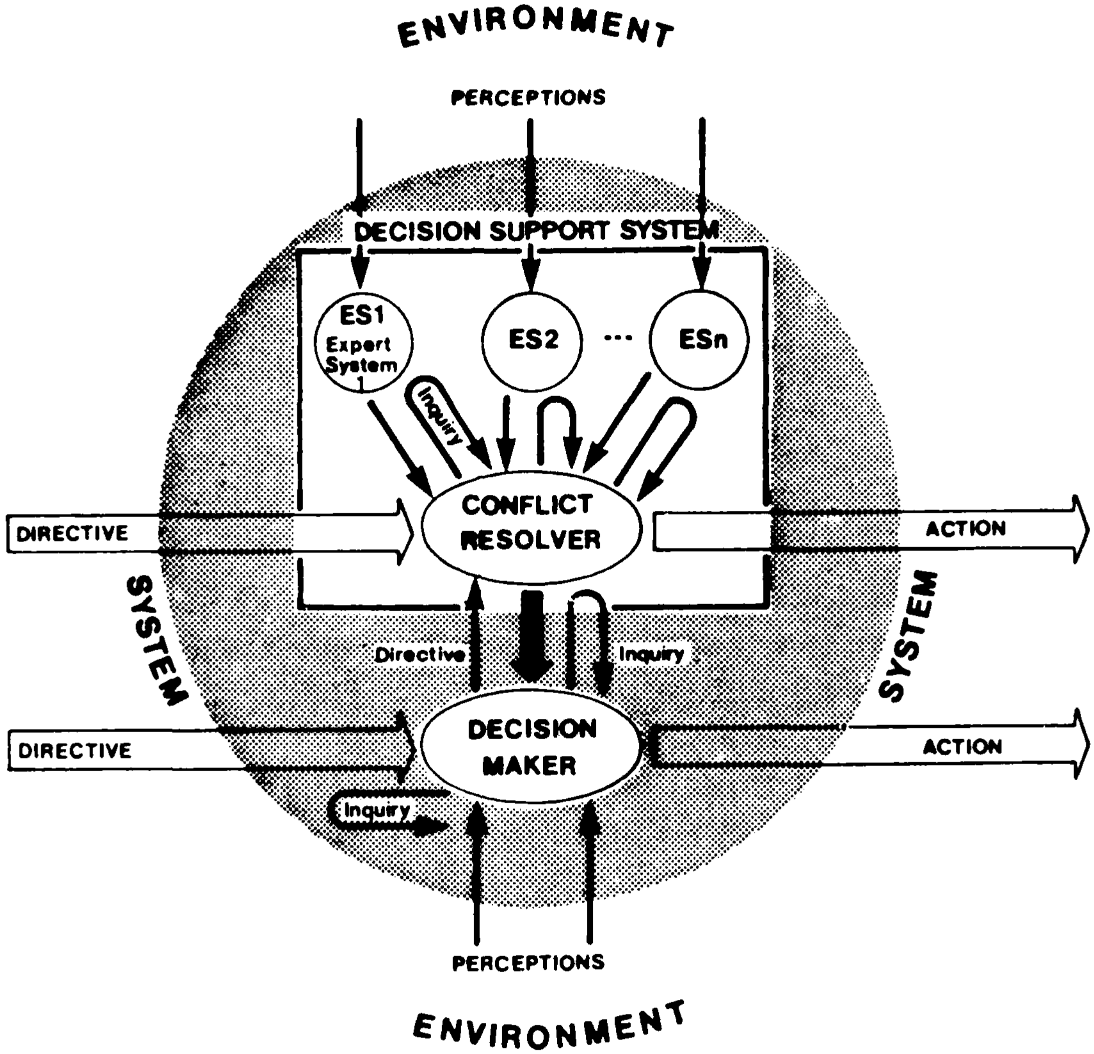


Figure 1: Problem Structure

The implementation of the query system between the ESs and the CR presented in this paper makes use of a "blackboard" through which the systems communicate. The blackboard is not a part of the general system problem, but rather a particular method of implementation that allows information to be shared. Consequently the blackboard does not appear in Figure 1.

The general procedure used by the CR is based on the Delphi method of consensus development, as presented in Section 2.5. In the DSS, each ES can revise its optimal recommendation (OR) based on the recommendations of the other ESs.

In the first step, each ES makes an OR on the basis of the data it has available. This OR is posted on the blackboard, and is available to the other ESs. Subsequently, each ES now has the OR of every other ES in addition to the original external data. This new information may change the OR for one or more ESs. This procedure is described below in Chapter 4.

It is assumed that the real world activities are slower than the decision making process of the DSS. Thus, the DSS can complete its analysis before changes in external data require the recommendation to be revised.

In the situation where each ES evaluates the possible recommendations according to a cardinal utility scale, it is possible to use mathematical principles to evaluate independently the eventual convergent overall recommendation. This permits the very rapid evaluation of the final recommendation without necessitating the interative reevaluation of all of the available information by each ES. This numerical approach was mathematically developed and experimentally implemented in this project.

3.2 Assumptions and Definitions

The Conflict Resolver must integrate the recommendations of the ESs on all decisions. Each individual decision is represented as an *action*, or binary *action variable*; the actions will be denoted $a_1, a_2, \cdots, a_n$, where

$$a_j = \begin{cases} 1 & \text{if action } j \text{ is selected} \\ 0 & \text{if action } j \text{ is not selected} \end{cases}$$

Note that the total number of available actions, n, is assumed to be finite. This means that all decisions must be represented in discrete form in the DSS.

The set of all conceivable decisions (which may include many that are infeasible) is then 2^n, where

$$\begin{aligned} 2^n &= \text{set of all sequences of } n \text{ 0's and 1's} \\ &= \text{set of all subsets of } N = \{1,2,\cdots,n\} \\ &= \text{Boolean (or Power) set of } N \end{aligned}$$

Of course, not every decision in 2^n need be feasible. Denote the *set of all feasible* *decisions* by $A \subseteq 2^n$. Note that, if $S \in A$, then S is a sequence of n 0's and 1's, or, equivalently, $S \subseteq N$.

Now, assume that there are m ESs (or other knowledge sources) labelled $ES1, \cdots, ESm$. Each ESs corresponds to a specific subset of the action variables on which it can make a recommendation. For $i = 1,2, \cdots, m$, the *decision domain* of ESi is $N_i \subseteq N$; this means that ESi is permitted to make a recommendation on the action variables $\{a_j: j \in N_i\}$. Note that $N_i \neq \phi$ is assumed.

A *feasible recommendation* for ESi is a subset $S_i \subseteq N_i$ such that, for some $S \in A$, $S_i = S \cap N_i$. Equivalently, a feasible recommendation for ESi can be represented as a sequence of $\#(N_i)$ 0's and 1's. The *set of all feasible recommendations for ESi* is then

$$A_i = \{S \cap N_i: S \in A\}$$

Note that $A_i \subseteq 2^{\#(N_i)}$.

Expert System i is assumed to have a *valuation* (or utility function)

$$v_i: A_i \rightarrow \mathbf{R}$$

The value of v_i depends explicitly on the values of the action variables in N_i, but v_i may also depend implicitly on the values of action variables outside N_i, as well as on external data. If $S_i \in A_i$, the value of $v_i(S_i)$ measures the extent to which the recommendation S_i meets the objectives of ESi, with higher values of $v_i(S_i)$ indicating greater success. An *optimal recommendation (OR) for ESi* is any $S_i^* \in A_i$ such that

$$v_i(S_i^*) \geq v_i(S_i), \forall S_i \in A_i$$

Later on, recommendations will be iterated using a revised valuation w_i, but the same definitions will apply.

Suppose that $S_i \in A_i$ is a feasible recommendation for each ESi, $i = 1,2, \cdots, m$. Then $(S_1, S_2, \cdots, S_m)$ is a system of recommendations. The system of recommendations $(S_1, S_2, \cdots, S_m)$ is *consistent* iff there exists $S \in A$ such that $S_i = S \cap N_i$ for each $i = 1,2, \cdots, m$.

A feasible decision $S^* \in A$ is an *optimal decision* iff, for each $i = 1,2, \cdots, m$, $S_i^* = S^* \cap N_i$ is an optimal recommendation for ESi. Note that if S^* is an optimal decision, then $(S_1^*, S_2^*, \cdots, S_m^*)$ is automatically a consistent system of recommendations. The objective of the iteration given below is to find a system of recommendations which is consistent and optimal with respect to valuations which are as close as possible to the original valuations.

3.3 System Components

The overall DSS has three main components:

1) Expert Systems (ESs). Each has its own knowledge base and inference engine, and has access to real world data. Each has a limited sphere of interest, expressed as N_i. For example, an ES that specializes in flight operations will not make recommendations having to do with armament systems. One important characteristic of the ES is that it can take into account new data as it becomes available. One component of this new data can be the recommendations of other ESs.

2) Blackboard. A blackboard permits different ESs to share conclusions and data. As each expert system makes an OR, it is written on the blackboard. Other associated or background data are written there also. The blackboard also contains a ranking, controlled by the CR, of the relative authority of each of the ESs at any point in time.

3) Conflict Resolver (CR). The CR selects from a various ORs the single comprehensive recommendation to pass on to the DM.

More details about these components are given in separate sections below.

Expert Systems

At any time it may be necessary to add or delete ESs. Also, the knowledge base of any ES may be expanded or changed. If each ES must know how to interpret every other ES, this could be impractical because of prohibitive time requirements.

The solution is to not require that each ES know the source of a particular OR. What ESi reads from the blackboard is that ESj has made recommendation S_j^* (of which only a subset may intersect with A_i), and that ESj has a certain level of authority in the ES rankings. This approach totally insulates each ES from any effects of adding or deleting another ES. Programming or reprogramming each ES only requires the capability to read and take into account these external recommendations.

In the implementation presented below, each ES reevaluates its recommendation using cardinal utilities and a numeric updating scheme. In practice, a scheme most suited to the actual design of the ESs would be chosen.

Blackboard

The blackboard is simply a place to keep data in a form accessible to all the ESs and the CR. As envisioned, it is dynamic and has no intelligence of its own.

Information to be retained on the blackboard includes:

1) The current recommendations, S_i, $i = 1, \cdots, m$. Each S_i is written by ESi only.

2) The weighting (ranking) of ESs, from most authoritative to least authoritative.

3) Special status reports. There may be specific conditions of interest to all ESs, such as combat, loss of certain capabilities, etc., that cannot easily be independently ascertained.

Conflict Resolver

The CR has several requirements:

1) It must extract S^* from the posted S_i^*. This is done by taking $S^* = \underset{i}{U} S_i^*$ if there is consistency, or selecting actions from the ORs of the ESs for each a_i according to the position of the ES in the rankings of authority.

2) It must maintain the ranking detailing the relative authority of the ESs. This is done by:

   a) reference to basic defaults

   b) medium term settings (eg. mission goals)

   c) short term settings (eg. pilot's instructions)

   The CR may have standard settings to correspond to different external environments, eg. combat, low altitude flight etc.

3) It must take care of various logical relationships among actions proposed in the S_i^*. There are three possible situations relating any two actions a_i and a_j:

   a) unrelated; eg. increase speed by 2 mph. and turn on map light.

   b) mutually exclusive. There are several subcases:

      i) a range of values, such as 800 mph vs. 900 mph,

      ii) opposite: turn left or turn right; open switch a or close switch a.

      iii) different but incompatible: 4g accelleration and eject at the same time.

   c) dependent, eg. a missile must be armed before it can be fired.

4) It must interact with the DM to present courses of action and accept queries and instructions.

4. MATHEMATICAL DEVELOPMENT OF THE CONFLICT RESOLVER

4.1 Notation and Definitions

Some additional assumptions and definitions, beyond those given in 3.2, are needed to define the CR methodology fully. As well, special notation must be introduced to express conveniently some ideas introduced in 3.2.

Each Expert System, *ESi*, makes a feasible recommendation $S_i \in A_i$. A complete system of recommendations is therefore

$$(S) = (S_1, S_2, \cdots, S_m) \in \prod_{i=1}^{m} A_i = XA$$

Note that $(S) \in XA$ need not be consistent, but, if (S) is consistent, then (S) corresponds to a unique feasible decision $S \in A$.

For $(S) = (S_1, S_2, \cdots, S_m) \in XA$ and $S_i' \in A_i$ for some i, define

$$(S_i, S_i') = (S_1, S_2, \cdots, S_{i-1}, S_i', S_{i+1}, \cdots, S_m)$$

In other words, (S_i, S_i') is the system of recommendations obtained from $(S) = (S_1, \cdots, S_i, \cdots, S_m)$ on replacing S_i by S_i'.

An important technical assumption, not discussed previously, concerns independence of ESs. Expert System i is *independent* iff, whenever $S^1 \in A$ and $S^2 \in A$, then there exists $S^3 \in A$ such that

$$S^3 \cap N_i = S^1 \cap N_i \text{ and } S^3 \cap (N - N_i) = S^2 \cap (N - N_i)$$

In other words, *ESi* is independent iff it is possible to take any feasible decision and alter it to match any other feasible decision inside N_i, while leaving it unchanged outside N_i. Thus, *ESi* is independent iff feasibility inside N_i does not depend on the actions selected outside N_i. An independent ES makes recommendations about any actions which can constitute necessary conditions for other actions on which it also makes recommendations.

Finally, assume that weights are assigned to ESs to fulfil two purposes which are described shortly. The *weight* for *ESi* is denoted W_i. It is assumed that $W_i > 0$ for $i = 1, 2, \cdots, m$, and that all weights are different.

The first purpose of the weights is to indicate the positions of the ESs in the (current) hierarchy. A greater weight will always be taken to indicate a higher position in the ordering. (This hierarchy is used for determining a temporary decision - see 4.2.) A second role for the weights is to measure the relative importance of agreement between two ESs. In the iteration (see 4.3), the relative worth of agreement between ESi and ESj on a common action is proportional to $W_i W_j$: Note that the weights may depend on external data, so that positions in the hierarchy can shift according to the circumstances.

4.2 Temporary Decision

When external data change significantly, or a decision is called for, each active ES makes a recommendation. These recommendations are then integrated quickly into a temporary (or provisional) feasible decision for posting on the blackboard. This posting is necessary because any ES's evaluation can depend on system actions outside its specific decision domain. For example, an ES low in the current hierarchy may make its recommendation so as to complement the recommendations of currently more important ESs.

The temporary decision procedure is simple and depends only on the current system of recommendations and the current hierarchy of ESs. The recommended choices of the highest ranked ES are fixed. Then, subject to feasibility, the recommendations of the next highest ES in the hierarchy, with respect to action variables not already set, are followed, subject to feasibility. This procedure is repeated for lower and lower ranking ESs until values have been chosen for all decision variables. In this way, a temporary decision, sufficient to determine each ES evaluation, is obtained. Note that, if the original system of recommendations is consistent, then the temporary decision is the unique feasible decision implied by the original system.

4.3 Conflict Resolver Algorithm

The algorithm used by the CR will now be described both formally and informally. The definitions given in 3.2 and 4.1 are assumed, and the proofs of theorems appear in the Appendix. The CR works by iteration - at each repetition of the iteration, it needs a temporary decision which will be assumed to be determined as in 4.2.

The CR uses an iterative procedure to pass from a system of optimal recommendations to reach (eventually) a nearby system of recommendations which is consistent. The iteration is modelled on the Delphi Procedure - each system of recommendations which arises in the iteration is optimal with respect to evaluations which depend on increasing amounts of consistency among recommendations.

To specify the iteration used by the CR, it is necessary to develop sophisticated methods of measuring the amount of consistency in a system of recommendations. Let $(S) \in XA$ and fix i and j. The *match-count* of ESi and ESj under (S) is

$$M_{ij}(S) = \#(S_i \cap S_j) + \#([N_i - S_i] \cap [N_j - S_j])$$

Thus, $M_{ij}(S)$ is the number of common actions on which ESi and ESj agree. Defining

$$n_{ij} = \#(N_i \cap N_j), \ \ n_i = \#(N_i)$$

provides some bounds on $M_{ij}(S)$:

Theorem 1: $0 \leq M_{ij}(S) \leq n_{ij}$. If $i = j$, $M_{ii}(S) = n_i$.

It is also possible to characterize consistency in terms of match-counts:

Theorem 2: $(S) \in XA$ is consistent iff $M_{ij}(S) = n_{ij}$ for all i,j.

Match-counts measure the consistency of a system of recommendations from the point of view of two specific ESs. A measure which adopts the viewpoint of a single ES will be defined now. Define the increment of (S) to ESi to be

$$Inc_i(S) = W_i \sum_{j=1}^{m} M_{ij}(S) W_j$$

where $(S) \in XA$. Of course, $Inc_i(S)$ is simply the weighted sum of all match-counts of (S) involving ESi. Note also that agreement on an additional action by ESi and ESj adds $W_i W_j$ to $Inc_i(S)$. It can be shown that

Theorem 3: $n_i W_i^2 \leq Inc_i(S) \leq \sum_{j=1}^{m} n_{ij} W_i W_j$.

Theorem 4: $(S) \in XA$ is consistent iff $Inc_i(S) = \sum_{j=1}^{m} n_{ij} W_i W_j$, for all i.

It is now straightforward to combine the consistency measures for single ESs into an overall consistency index. If $(S) \in XA$, define the *consistency index* at (S) to be

$$Con(S) = \sum_{i=1}^{m} Inc_i(S)$$

It then follows that

Theorem 5: $Conmin \leq Con(S) \leq Conmax$, where

$$Conmin = \sum_{i=1}^{m} n_i W_i^2, \ Conmax = \sum_{i=1}^{m} \sum_{j=1}^{m} n_{ij} W_i W_j$$

Theorem 6: $(S) \in XA$ is consistent iff $Con(S) = Conmax$.

The operation of the CR iteration will now be described. Assume a current system of recommendations $(S^*) = (S_1^*, S_2^*, \cdots, S_m^*) \in XA$ such that, for each i, S_i^* is an optimal recommendation for ESi under evaluation $v_i(\cdot)$. A single application of the iteration algorithm will produce a revision to S^* which makes it more consistent according to the consistency index $Con(S^*)$, and which preserves the optimality property, though with respect to a slightly altered set of evaluations. Evaluations are always made in light of the current temporary decision; as well, a minimal bonus for consistency is introduced at each step. It is the two latter properties which are analogous to the Delphi Procedure.

CR Algorithm Step 1:

For each ESi, for each $S_i \in A_i$, define the new evaluation

$$w_i(S_i,t) = v_i(S_i) + t Inc_i(S_i^*, S_i)$$

where $t \geq 0$ is to be specified. (Here w_i is the revised evaluation and t measures the bonus for consistency.) For ESi, the minimum effective value of t is $t_{min}(i)$, defined as follows:

If $Inc_i(S^*) \geq Inc_i(S_i^*, S_i)$ for all $S_i \in A_i$, $t_{min}(i) = \infty$. Otherwise let $A_i' = \{S_i \in A_i \colon Inc_i(S_i^*, S_i) > Inc_i(S^*)\}$ and

$$t_{min}(i) = \min\left\{ \frac{v_i(S_i^*) - v_i(S_i)}{Inc_i(S_i^*, S_i) - Inc_i(S^*)} \colon S_i \in A_i' \right\}$$

(Note that, if $t \geq t_{min}(i)$, then some recommendation other than S_i^* rates as high as S_i^* under the evaluation $w_i(\cdot,t)$.)

CR Algorithm Step 2:

Set $t_{min} = \min\{t_{min}(i) \colon i = 1,2,\cdots,m\}$. If $t_{min} = \infty$, the iteration stops. Otherwise, identify i so that $t_{min} = t_{min}(i) < \infty$. For ESi, there exists $\bar{S}_i \in A_i'$ such that $w_i(\bar{S}_i, t_{min}) = w_i(S_i^*, t_{min})$. Identify $\bar{S}_i$.

CR Algorithm Step 3:

Replace S_i^* by $\bar{S}_i$, so that the current system of recommendations changes $(S_1^*, \cdots, S_i^*, \cdots, S_m^*) \rightarrow (S_1^*, \cdots, S_{i-1}^*, \bar{S}_i, S_{i+1}^*, \cdots, S_m^*)$. For each $j = 1,2,\cdots,m$, replace $v_i(\cdot)$ by $w_i(\cdot, t_{min})$. Calculate a new temporary decision (see 4.2), and return to Step 1.

The demonstration that the iteration specified above always reaches a consistent system of recommendations in a finite number of steps, and stops when and only when it attains consistency, is divided into three parts:

Theorem 7: The CR Algorithm stops if (S^*) is consistent.

Theorem 8: The CR Algorithm does not stop if (S^*) is not consistent.

Theorem 9: The CR Algorithm stops after finitely many steps.

In conclusion, a few comments on the CR Algorithm are appropriate. Little is known at present about the speed of convergence of the iteration, or about how the speed might be affected by certain possible modifications. Also, the implications of the failure of certain assumptions, such as the independence assumption of 4.1, are unexplored. Finally, it is possible for the algorithm to branch - to include arbitrary decisions - although this is technically an event of low probability. Nonetheless, the implications of branching are not understood at the present time. These and other issues deserve further intensive study beyond what is reported here.

5. IMPLEMENTATION

Two experimental implementations of a CR were made. Both assumed that the basis for the ESs decisions devolved to a comparison of the utilities of various recommendations. The principles developed for this experimental implementation can be applied to various actual decision selection schemes.

The first implementation involved three ESs, each of which had interest in three action variables out of five available. There are thus $2^5 = 32$ feasible decisions, and each ES has a valuation on $2^3 = 8$ feasible recommendations.

In this first implementation, the Supercalc3 spreadsheet program (similar to the popular Lotus123 package) was employed. Supercalc has a manual updating feature that is normally used to recalculate the effect that changes in entries have on calculations that involve them. In the CR implementation, this updating facility is used to control the steps of information exchange and optimal recommendation revision among the ESs. A composite screen display from the Supercalc program is shown in Table 2.

At each stage of the conflict resolution process, every ES revises its valuation of its feasible recommendations on the basis of the current optimal recommendation of the other two ESs. In this implementation, a variable bonus (in terms of utility) is given as to how many of the actions in a particular feasible recommendation match the appropriate actions in the optimal recommendations of the other ESs in the previous step. This bonus is indicated in Table 2 as "higher ES boost", applying for ESs higher in rank, and "lower ES boost" for those lower. In this manner, the utility of the mutually agreeable feasible recommendations increases. Eventually, an overall consensus is reached. This is shown in Table 2 as the column labelled "next payoff", where strategies 001, 010 and 100 are best for ES1, ES2 and ES3, respectively. As shown in the upper left corner of the table, these are mutually consistent, and the overall S^* of 00100 is achieved.

This implementation was very valuable in studying the performance of the scheme. In particular, the conditions under which the system failed to converge were examined. It was observed that if the step size was too large, cycling could occur. This is because the bonus utility given to two or more ESs can change their optimal recommendation simultaneously. A cycle happens if the bonus causes two or more ESs to alternate among optimal recommendations.

One way to solve this problem is to make the bonus small. The problem with this approach is that the required computations approach infinity as the bonus approaches 0. If the system is to run in real time, it must be fast.

A particularly efficient approach is to calculate the minimum bonus amount necessary to cause one ES to change its optimal recommendation. With this approach, each step in the resolution procedure actually causes a change in optimal recommendation, rather than just a change in the utility evaluation.

A FORTRAN program was written to implement this approach. A very simple model of two ESs with a total of two action variables is used, although the program is written in such a manner that it can be easily expanded for more ESs and more action variables. In the current implementation, each ES is concerned with both action variables.

Extensive testing shows that the approach converges to the correct optimal decision in two steps. It is should be very efficient for large systems, especially since there are many obvious short cuts evident in the solution procedures. Output from the FORTRAN program is shown in Table 3.

Utility data for each ES is read from a file. The input values in Table 3 indicate that ES1 evaluates the worth of the situation where ES1 takes action 1 and ES2 takes action 2 at 11, where ES1 takes action 1 while ES2 does not take action 2 at 6, where ES1 does not take action 1 while ES2 does take action 2 at 10 and where ES1 does not take action 1 while ES2 does not take action 2 at 5. Similarly, ES2 evaluates the situation where both ESs take their action at 3.

The weights and t-values are used as described in Section 4.2 to determine the next updating of the utilities. The results are shown in the middle of Table 3. However, there is not yet convergence, so the process must be repeated. Near the bottom of Table 3, it can be seen that agreement between the two ESs has been achieved, at the situation where ES1 does not take action 1 while ES2 does take action 2. As presented in Section 4, convergence will always occur; it has been observed that in the simple model of Table 3, it always occurs in two steps.

Table 2. Supercalc screen display for Implementation No. 1.

Conflict resolver test

higher ES boost 2
lower ES boost: 1 Go: 1

Summary
s1*: 0 0 1
s2*: 0 1 0
s3*: 1 0 0
convergence
S*: 0 0 1 0 0

	a1	a2	a3	a4	a5	init. payoff	next. payoff	curr. bonus
ES1	0	0	0			70	76	1
	1	0	0			30	55	0
	0	1	0			80	86	1
	0	0	1			90	122	3
	1	1	0			30	55	0
	1	0	1			40	91	2
	0	1	1			30	62	3
	1	1	1			10	61	2
s1*	0	0	1					
ES2	0		0		0	40	91	3
	1		0		0	20	33	1
	0		1		0	50	139	5
	0		0		1	80	124	2
	1		1		0	40	91	3
	1		0		1	60	66	0
	0		1		1	10	92	4
	1		1		1	90	134	2
s2*	0		1		0			
ES3			0	0	0	30	30	2
			1	0	0	70	146	6
			0	1	0	20	20	2
			0	0	1	50	88	0
			1	1	0	40	116	6
			1	0	1	30	144	4
			0	1	1	80	118	0
			1	1	1	10	124	4
s3*			1	0	0			

Table 3. Output listing for Implementation No. 2.

```
The input file name is: cr1.dat
The output file name is: cr1.out

The input values are:
             for ES1:      11      6
                           10      5
             for ES2:       3      6
                            7     10

The maximums are located:
             for ES1: (1,1)
             for ES2: (2,2)

The weights are:
             for ES1:                .00              1.00
                                    1.00              2.00
             for ES2:               2.00              1.00
                                    1.00               .00

The t-values are:
             for ES1:           99999.00              5.00
                                    1.00              3.00
             for ES2:               3.50              4.00
                                    3.00          99999.00

The minimum t-value is located at: (1,2,1)

The current payoffs are:
             for ES1:      11      7
                           11      7
             for ES2:       5      7
                            8     10


The maximums are located:
             for ES1: (2,1)
             for ES2: (2,2)

The weights are:
             for ES1:                .00              1.00
                                    1.00              2.00
             for ES2:               1.00               .00
                                    2.00              1.00

The t-values are:
             for ES1:           99999.00          99999.00
                                99999.00              4.00
             for ES2:           99999.00          99999.00
                                    2.00          99999.00

The minimum t-value is located at: (2,2,1)

The current payoffs are:
             for ES1:      11      9
                           13     11

             for ES2:       7      7
                           12     12


The maximums are located:
             for ES1: (2,1)
             for ES2: (2,1)


*** CONVERGENCE ***  at decision: (2,1)
```

6. CONCLUSIONS AND RECOMMENDATIONS

Within this report, a comprehensive CR was developed for mediating disputes arising among competitive ESs. In addition to ESs and other knowledge sources, the CR forms a key component of a flexible decision support system, designed for operating a complex system. To determine iteratively an overall recommendation from the conflicting suggestions of competing ESs, the CR follows a Delphi approach to mediation. Besides the design of the mathematical framework for the comprehensive CR, a mathematical proof that it always converges to a consistent recommendation has been provided here.

The foregoing accomplishments constitute a significant and important step in the development of ES methodology. As a matter of fact, due to the great demand for ESs in the military and many other fields, it was probably inevitable that some research group would eventually design a flexible CR. Nonetheless, in order to capitalize upon the current leadership in this area as well as making the implementation of the CR a practical reality, much work remains to be done. Specifically, recommendations for future work include:

1. Further theoretical work is required as explained in Section 4.3.
2. Algorithms that will allow the CR to be implemented conveniently in practice must be developed.
3. Flexible computer programs which implement the CR on a microcomputer, or possibly a main frame computer, should be developed for research and testing purposes.
4. Extensive simulation studies are needed to ascertain the performance of the CR over a wide range of possible scenarios and operating conditions.
5. Case studies are required to calibrate and test the CR under specific situations that closely reflect reality. In other words, the CR should be also tested under field conditions that may arise within a specific context, such as the Pilot's Associate.
6. Further work should be done to develop, both theoretically and practically, other components of the decision support system shown in Figure 1. For example, a blackboard architecture may have to be designed to allow the ESs, CR and decision maker to communicate efficiently with one another.

7. A conflict resolver could be designed for use as an internal component of a given ES in order for that ES to decide upon its own recommendation when confronted with conflicting information.

8. Because the use of a comprehensive decision support system (see Figure 1), such as a Pilot's Associate, will allow a decision maker to perform in a more optimal fashion, there are tremendous benefits to be gained. Comprehensive studies can be carried out to determine benefits, such as economic, social and military, that can be obtained when a comprehensive decision support system is implemented. For example, the use of a Pilot's Associate will allow the effective integration of military hardware with the human capacities of the pilot to optimize the performance of the overall system.

REFERENCES

Anderson, J.R., "Cognitive Psychology" (Correspondent's Report), *Artificial Intelligence*, Vol. 23, pp. 1-11, 1984.

Anderson, B.M., McNulty, C., and Lystad, G.S., "Expert Systems for Aiding Combat Pilots", *First Annual Aerospace Applications of Artificial Intelligence (AAAI) Conference Proceedings*, Dayton, Ohio, pp. 54-60, Sept. 16-19, 1985.

Barr, A., and Feigenbaum, E.A., *The Handbook of Artificial Intelligence, Vol. 3,* William Kaufman, Los Altos, California, 1982.

Bobrow, D.G., and Collins, A. (Editors), *Representation and Understanding,* Academic Press, New York, 1975.

Boden, M., *Artificial Intelligence and Natural Man,* Basic Books, New York, 1977.

Bonczek, R.H., Holsapple, C.W., and Whinston, A.B., "The Evolving Roles of Models in Decision Support Systems", *Decision Sciences,* Vol. 11, No. 2, pp. 339-356, 1980.

Bonczek, R.H., Holsapple, C.W., and Whinston, A.B., *Foundations of Decision Support Systems,* Academic Press, New York, 1981.

Buchanan, B.G., "New Research on Expert Systems", in *Machine Intelligence 10,* edited by J.E. Hayes, D. Michie and Y.H. Pao, Ellis Horwood, Chichester, England, and Halstead, New York, pp. 269-299, 1982.

Buchanan, B.G., "Artificial Intelligence: Toward Machines that Think", in *Encyclopedia Brittanica, 1985 Yearbook of Science and the Future,* 1985.

Buchanan, B.G., and Shortliffe, E.H., *Rule-Based Expert Programs: The MYCIN Experiments of the Stanford Heuristic Programming Project,* Addison-Wesley, Reading, Massachusetts, 1984.

Charniak, E., Riesbeck, C.K., and McDermott, D.V., *Artificial Intelligence Programming,* Erlbaum, Hillsdale, New Jersey, 1980.

Dalkey, N., *The Delphi Method - An Experimental Study of Group Opinion,* Technical Report Number RM-3855-PR, The Rand Corporation, 1969.

Davis, R., and Lenant, D.B., *Knowledge-Based Systems in Artificial Intelligence,* McGraw-Hill, New York, 1982.

Dutta, A., and Jain, H.K., "A DSS for Distributed Computer System Design in the Presence of Multiple Conflicting Objectives", *Decision Support Systems,* Vol. 1, pp. 233-246, 1985.

Feigenbaum, E.A., and Feldman, J. (Editors), *Computers and Thought,* McGraw-Hill, New York, 1963.

Findler, N.V., *Associative Networks - Representation and Use of Knowledge by Computers,* Academic Press, New York, 1979.

Fraser, N.M., and Hipel, K.W., *Conflict Analysis: Models and Resolutions,* North-Holland, New York, 1984.

Ginzberg, M.J., Reitman, W., and Stohr, E.A. (Editors), "Decision Support Systems", *Proceedings of the New York University Symposium on Decision Support Systems,* May 21-22, 1981, North-Holland, Amsterdam, 1982.

Gorry, G.A., and Scott Morton, M.S., "A Framework for Management Information Systems", *Sloan Management Review,* Vol. 13, No. 1, pp. 55-70, 1971.

Hayes-Roth, B., "A Blackboard Architecture for Control", *Artificial Intelligence,* Vol. 26, pp. 251-321, 1985.

Hayes-Roth, F., Waterman, D.A., and Lenant, D.B. (Editors), *Building Expert Systems,* Addison-Wesley, Reading, Massachusetts, 1983.

Howard, N., *Paradoxes of Rationality, Theory of Metagames and Political Behaviour,* M.I.T. Press, Cambridge, Massachusetts, 1971.

Hoyland, C.M., Evers, K.H., and Snyder, D.E., "Incorporating Human Operator Considerations into Existing Weapon System Analysis and Quantification Capabilities", *Proceedings of the 1985 National Aerospace and Electronics Conference (NAECON),* Dayton, Ohio, pp. 1-6, May, 1985.

Jarke, M., "Knowledge Sharing and Negotiation Support in Multiperson Decision Support Systems", *Decision Support Systems,* Vol. 2, pp. 93-102, 1986.

Keen, P.G.W., "Adaptive Design for Decision Support Systems", *Data Base,* Vol. 12, Nos. 1 and 2, 1980.

Keen, P.G.W., and Scott Morton, M.S., *Decision Support Systems, An Organizational Perspective,* Addison-Wesley, Reading, Massachusetts, 1978.

Kilgour, D.M., Hipel, K.W., and Fraser, N.M., "Solution Concepts in Non-Cooperative Games", *Large Scale Systems,* Vol. 6, No. 1, pp. 49-71, 1984.

Klein, H.K., and Hirscheim, R., "Fundamental Issues of Decision Support Systems: A Consequentialist Perspective", *Decision Support Systems,* Vol. 1, pp. 5-24, 1985.

Landry, M., Pascot, D., and Briolat, D., "Can DSS Evolve Without Changing Our View of the Concept of "Problem"?", *Decision Support Systems,* Vol. 1, pp. 25-36, 1985.

Luce, R.D., and Raiffa, H., *Games and Decisions,* John Wiley, New York, 1957.

McCourduck, P., *Machines Who Think,* Freeman, San Francisco, 1979.

McNeese, M.D., *Humane Intelligence: A Human Factors Perspective for Developing Intelligent Cockpits,* Proceedings of the 1986 National Aerospace and Electronics Conference (NAECON), Dayton, Ohio, pp. 941-948, May, 1986.

Mishkoff, H.C., *Understanding Artificial Intelligence*, Texas Instruments, Dallas, Texas, 1985.

Moore, J.H., and Chang, M.G., "Design of Decision Support Systems", *Data Base*, Vol. 12, Nos. 1 and 2, pp. 8-14, 1980.

Negoita, C.V., *Expert Systems and Fuzzy Systems,* Benjamin/Cummings, Menlo Park, California, 1985.

Nilsson, N.J., *Principles of Artificial Intelligence,* Tioga, Palo Alto, California, 1980.

Pylyshyn, Z.W., "Literature for Cognitive Psychology", *Artificial Intelligence,* Vol. 19, pp. 251-255, 1982.

Quade, E.S., "Predicting the Consequences: Models and Modelling", in *Handbook of Systems Analysis,* edited by H.J. Miser and E.S. Quade, North-Holland, New York, Ch. 7, 1985.

Rich, E., *Artificial Intelligence*, McGraw-Hill, New York, 1983.

Rumelhart, D.E., McClelland, J.L., and other members of the Parallel Distributed Processing Research Group, *Parallel Distributed Processing, Explorations in the Microstructures of Cognition, Vol. 1: Foundations, Vol. 2: Psychological and Biological Models,* M.I.T. Press, Cambridge, Massachusetts, 1986.

Sage, A.P., and Rouse, W.B., "Aiding the Human Decisionmaker through the Knowledge-Based Sciences", *IEEE Transactions on Systems, Man and Cybernetics,* Vol. SMC-16, No. 4, pp. 511-521, 1986.

Schank, R.C., and Colby, K.M. (Editors), *Computer Models of Thought and Language,* Freeman, San Francisco, 1973.

Sell, P.S., *Expert Systems - A Practical Introduction,* MacMillan, 99 pp., 1985.

Sen, A., and Biswas, G., "Decision Support Systems: An Expert Systems Approach", *Decision Support Systems,* Vol. 1, pp. 197-204, 1985.

Singh, M.G., and Cook, R., "A New Class of Intelligent Knowledge-Based Systems with an Optimization-Based Inference Engine", *Decision Support Systems,* Vol. 1, pp. 299-312, 1985.

Smith, E.E., "Cognitive Psychology" (Correspondent's Report), *Artificial Intelligence*, Vol. 25, pp. 247-253, 1985.

Sol, H.K., "Aggregating Data for Decision Support", *Decision Support Systems*, Vol. 1, pp. 111-121, 1985.

Sprague, R.H., Jr., "A Framework for the Development of Decision Support Systems", *MIS Quarterly*, Vol. 4, No. 4, pp. 1-26, 1980.

Sutherland, J.W., "Assessing the Artificial Intelligence Contribution to Decision Technology", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-16, No. 1, pp. 3-20, 1986.

Thierauf, R.J., *Decision Support Systems for Efficient Planning and Control - A Case Study Approach*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

Von Newmann, J., and Morgenstern, O., *Theory of Games and Economic Behaviour, Third Edition*, Princeton University Press, Princeton, New Jersey, 1953.

Winston, P.H., *Artificial Intelligence, Second Edition*, Addison-Wesley, Reading, Massachusetts, 1984.

Wortman, D.B., Duket, S.D., Seifert, D.J., Hann, R.L., and Chubb, G.P., *Simulation Using SAINT: A User-Oriented Manual*, Technical Report Number AMRL-TR-77-61(ADA 058-671), Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, 1977.

APPENDIX

This Appendix contains the proofs of theorems in 4.3. All formal definitions and assumptions given in 3.2, 4.1, and 4.3 are assumed and will not be repeated here. This Appendix should be read in conjunction with 4.3.

Proof of Theorem 1:

Because $S_i \subseteq N_i$ and $S_j \subseteq N_j$, it follows that

$$M_{ij}(S) = \#(N_i \cap N_j \cap S_i \cap S_j) + \#(N_i \cap S_i^c \cap N_j \cap S_j^c)$$

$$= \#(N_i \cap N_j \cap [(S_i \cap S_j) \cup (S_i^c \cap S_j^c)])$$

$$\leq \#(N_i \cap N_j) = n_{ij}$$

Also, it is obvious that $M_{ij}(S) \geq 0$ and that $M_{ij}(S) = n_i$. //

Proof of Theorem 2:

If $(S) = (S_1, S_2, \cdots, S_m)$ is consistent, then there exists $\bar{S} \in A$ such that $S_i = \bar{S} \cap N_i$ and $S_j = \bar{S} \cap N_j$. Therefore,

$$M_{ij}(S) = \#([\bar{S} \cap N_i] \cap [\bar{S} \cap N_j]) + \#([N_i - \bar{S}] \cap [N_j - \bar{S}])$$

$$= \#(N_i \cap N_j \cap \bar{S}) + \#(N_i \cap N_j \cap \bar{S}^c)$$

$$= \#(N_i \cap N_j) = n_{ij}$$

Now suppose that $M_{ij}(S) < n_{ij}$ for some i and j. Then, without loss of generality, there exists $k \in S_i \subseteq N_i$ such that $k \in N_j - S_j$. Now assume that $\bar{S} \in A$. If $k \in \bar{S}$, then $S_j \neq \bar{S} \cap N_j$, whereas, if $k \notin \bar{S}$, then $S_i \neq \bar{S} \cap N_i$. This shows that $(S_1, \cdots, S_m)$ cannot be consistent. //

Proof of Theorem 3:

Follows easily from Theorem 1. //

Proof of Theorem 4:

Follows easily from Theorem 2. //

Proof of Theorem 5:

Follows easily from Theorem 3. //

Proof of Theorem 6:

Follows easily from Theorem 4. //

Proof of Theorem 7:

The iteration stops if $t_{\min} = \infty$. Now $t_{\min} = \infty$ iff $t_{\min}(i) = \infty$ for $i = 1,2,\cdots,n$.

If (S^*) is consistent, then by Theorem 4, $Inc_i(S^*) = \sum_{j=1}^{m} n_{ij}W_iW_j$ for $i = 1,2,\cdots,m$. By Theorem 3, $Inc_i(S_i^*,S_i) \leq \sum_{j=1}^{m} n_{ij}W_iW_j$ for all i and all $S_i \in A_i$. It follows that

$$Inc_i(S^*) \geq Inc_i(S_i^*,S_i)$$

for all $S_i \in A$ and all i, so that $t_{\min}(s) = \infty$ for all i. As noted above, this implies that the iteration stops. //

Proof of Theorem 8:

Assume that S^* is not consistent. It follows from Theorem 2 that there exists i,j, $1 \leq i,j \leq m$, $i \neq j$, and $k \in N$ such that $k \in S_i^* \subseteq N_i$ and $k \in N_j - S_j^*$. Let $T_1 = \{h: k \in S_h^*\}$ and $T_2 = \{h: k \in N_h - S_h^*\}$. Then $T_1 \neq \phi$ and $T_2 \neq \phi$.

If $\sum_{h \in T_2} W_h \geq \sum_{h \in T_1} W_h$, choose any $i \in T_1$ and define $S_i = S_i^* - \{k\}$. Then

$$Inc_i(S_i^*,S_i) - Inc_i(S^*) = \sum_{h=1}^{m} M_{ih}(S_i^*,S_i)W_iW_h - \sum_{h=1}^{m} M_{ih}(S^*)W_iW_h$$

Now if $h \in T_2$, $k \in S_i^*$, $k \notin S_i$, and $k \notin S_h^*$ so that $M_i(S_i^*,S_i) - M_{ih}(S^*) = 1$, whereas if $h \in T_1 - i$, $k \in S_i^*$, $k \notin S_i$, and $k \in S_h^*$ so that $M_{ih}(S_i^*,S_i) - M_{ih}(S^*) = -1$. By Theorem 1, $M_{ii}(S_i^*,S_i) = M_{ii}(S^*) = n_i$. It follows that

$$Inc_i(S_i^*,S_{ii}) - Inc(S^*) = W_i[\sum_{h \in T_2} W_h - \sum_{h \in T_1 - i} W_h] > 0$$

Therefore, $t_{\min}(i) < \infty$ so that the iteration does not stop, as noted in the proof of Theorem 7.

If $\sum_{h \in T_2} W_h < \sum_{h \in T_1} W_h$, choose any $j \in T_2$ and define $S_j = S_j^* \cup \{k\}$. Then the proof that

$$Inc_j(S_j^*,S_j) - Inc_j(S^*) > 0$$

is analogous. //

Proof of Theorem 9:

The proof is accomplished by showing that, if $\bar{S}_i$ is obtained as in CR Algorithm Step 2, then $Con(S_i^*,\bar{S}_i) > Con(S^*)$. This demonstrates that the algorithm always acts to make the current system of optimal recommendations more consistent according to the index $Con(S^*)$. Furthermore, it will be proven that there exists $\epsilon > 0$ such that $Con(S_i^*,\bar{S}_i) - Con(S^*) > \epsilon$, where ϵ depends only on the weights assigned to the ESs. Combined with Theorem 6, this implies that the system of optimal recommendations will converge to a consistent optimal decision after finitely many iterations.

If $\bar{S}_i$ is as selected in Step 2,

$$Con(S_i^*,\bar{S}_i) - Con(S^*) = \sum_{j=1}^{m} Inc_j(S_i^*,\bar{S}_i) - \sum_{j=1}^{m} Inc_j(S^*)$$

$$= \sum_{j \neq i} \left[\sum_{k \neq i} M_{jk}(S_i^*,\bar{S}_i) W_j W_k + M_{ji}(S_i^*,\bar{S}_i) W_j W_i \right]$$

$$+ \sum_{k} M_{ij}(S_i^*,\bar{S}_i) W_i W_k - \sum_{j} \sum_{k} M_{jk}(S^*) W_j W_k$$

Now if $j \neq i$ and $k \neq i$, $M_{jk}(S_i^*,\bar{S}_i) = M_{jk}(S^*)$. By Theorem 1, $M_{ii}(S_i^*,\bar{S}_i) = M_{ii}(S^*) = n_i$. Also $M_{ij}(S_i^*,\bar{S}_i) = M_{ji}(S_i^*,\bar{S}_i)$ and $M_{ij}(S^*) = M_{ji}(S^*)$. These observations imply that

$$Con(S_i^*,\bar{S}_i) - Con(S^*) = 2\sum_{j \neq i} M_{ij}(S_i^*,\bar{S}_i) W_i W_j - 2\sum_{j \neq i} M_{ij}(S^*) W_i W_j$$

$$= 2\left[\sum_{j} M_{ij}(S_i^*,\bar{S}_i) W_i W_j - \sum_{j} M_{ij}(S^*) W_i W_j \right]$$

$$= 2[Inc_i(S_i^*,\bar{S}_i) - Inc_i(S^*)] > 0$$

Now assume that $ES1, ES2, \cdots, ESm$ are fixed so that $N_1, N_2, \cdots, N_m$ are fixed, and also the weights $W_1, W_2, \cdots, W_m$ are fixed. Then, for each i and j, there are a finite number of possible values of $M_{ij}(S)$ and, therefore, there are a finite number of possible values of $Inc_i(S) = \sum_j M_{ij}(S) W_i W_j$. Let ϵ_i be the minimum difference between any two (distinct) values of $Inc_i(S)$, and let $\epsilon = \min_i \epsilon_i$. Then $Con(S_i^*,\bar{S}_i) - Con(S^*) > 2\epsilon$, and this fact completes the proof, as noted above. //

END
8-87
DTIC